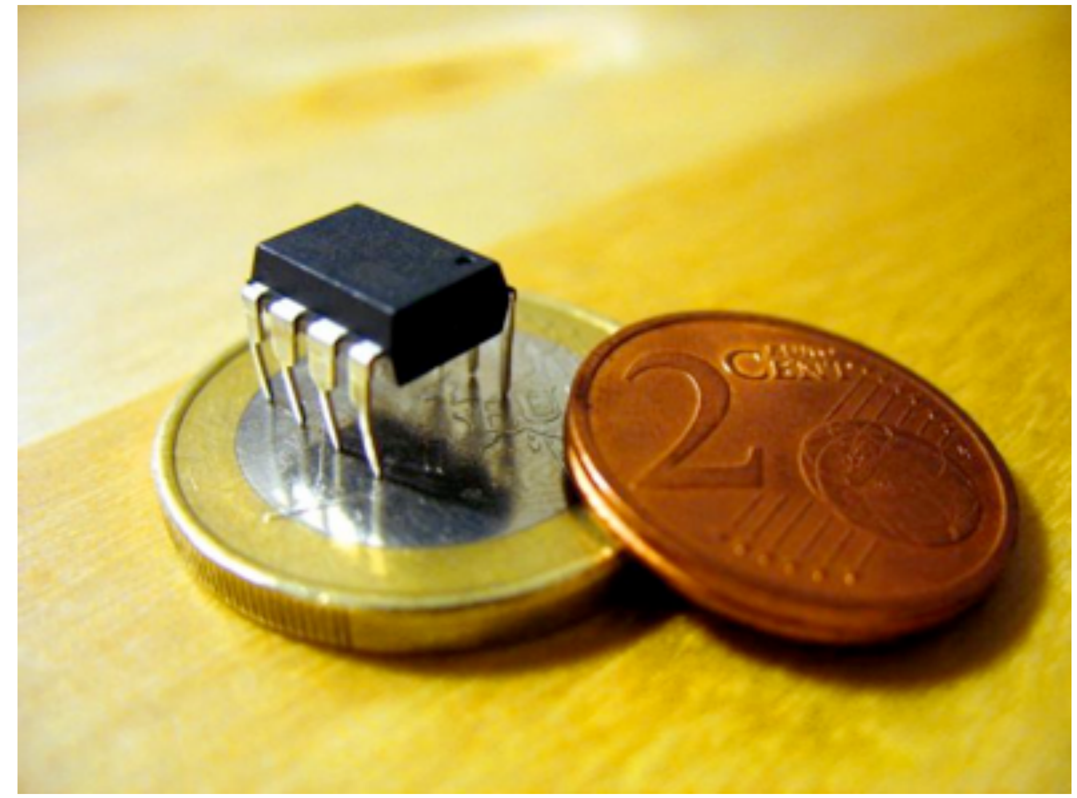
Imperial College London

Who needs IEEE double precision floating-point?

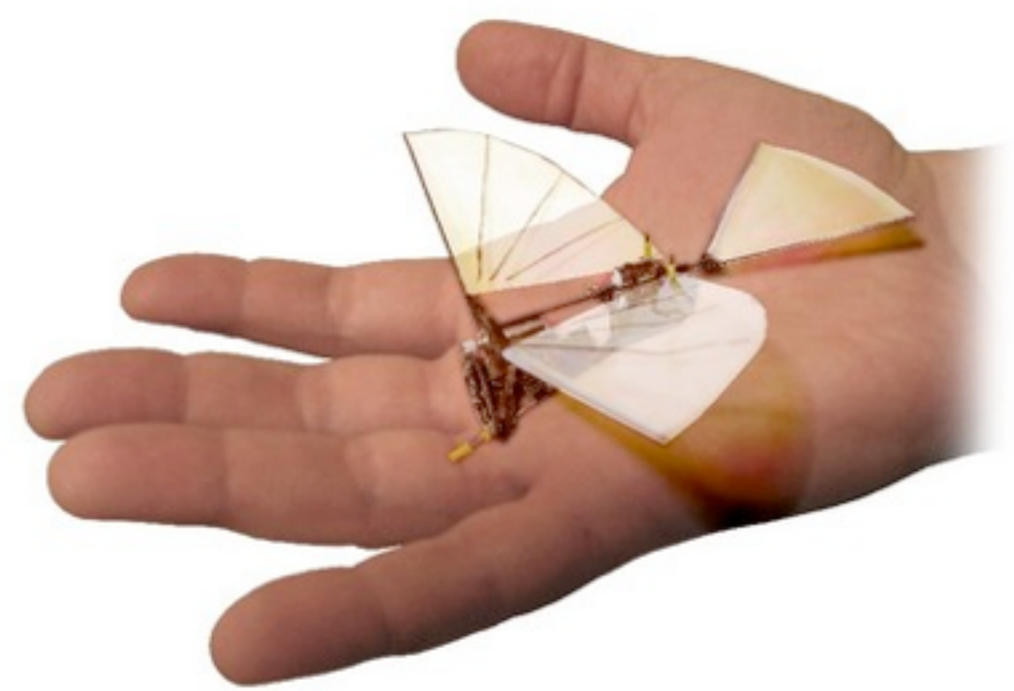
Eric Kerrigan, Juan Jerez, Stefano Longo, George Constantinides
Department of Electrical and Electronic Engineering
Department of Aeronautics

We Need **New** Ideas And We Need It **Soon**

Can you get a nonlinear model-based predictive controller to do aerobatics of a micro-UAV using a:



- 2 cent, 1 nW microcontroller
- in 5/10/20/50/100 years' time?

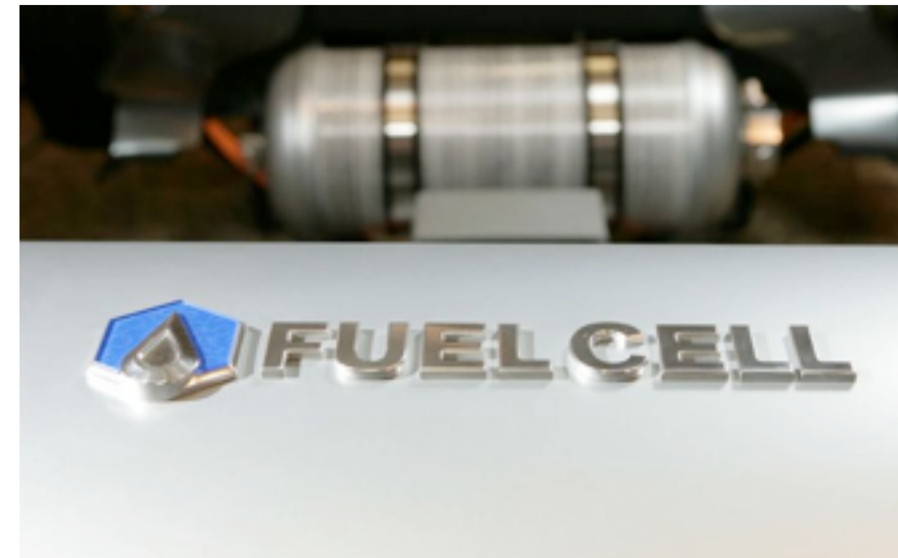


Old Ideas Will Probably **Never** Work

NVIDIA
TESLA™
PERSONAL
SUPERCOMPUTER

Applications for Embedded Optimization (Optimal Control / Estimation / DSP)



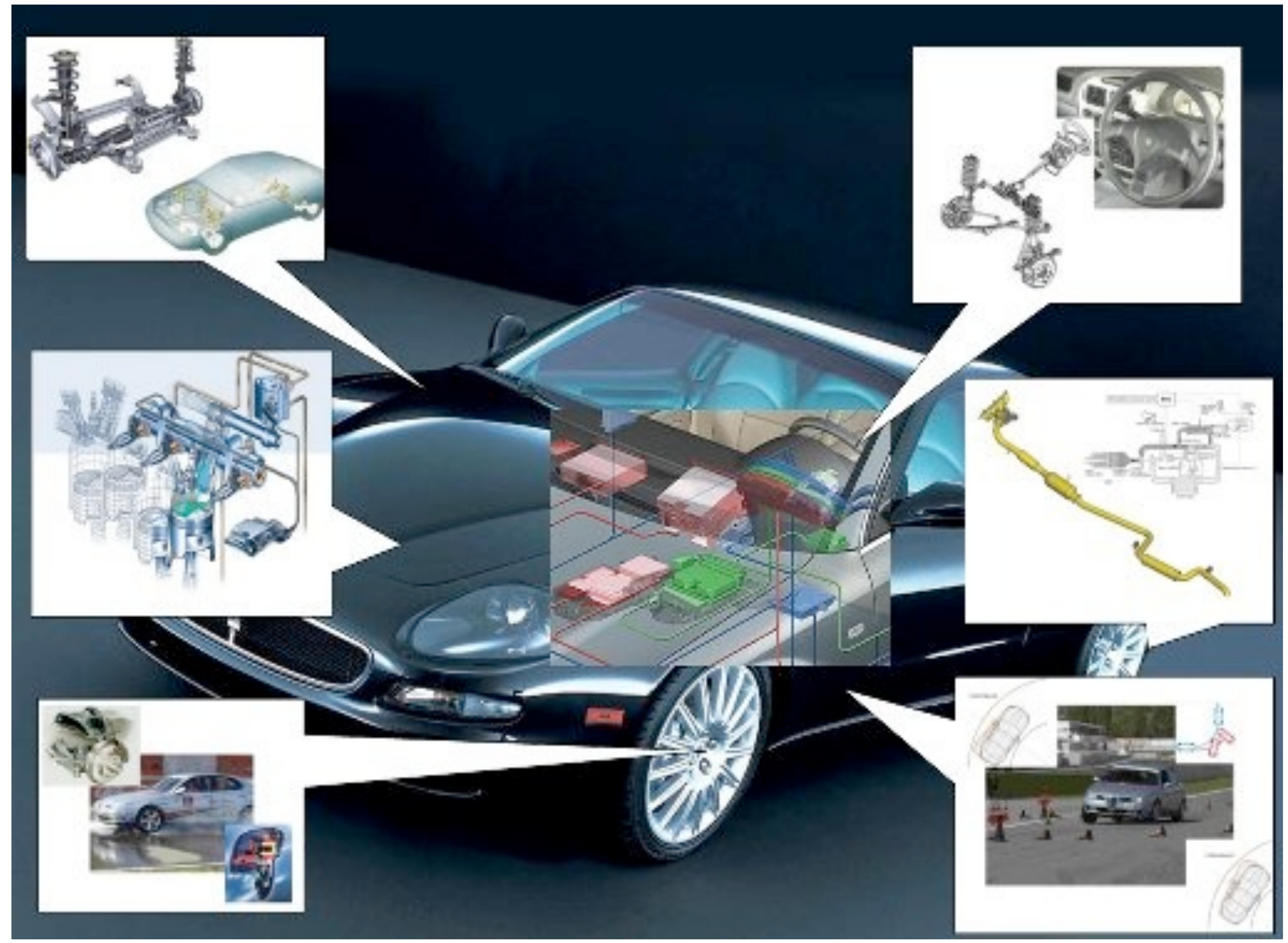






Challenges for Embedded Systems

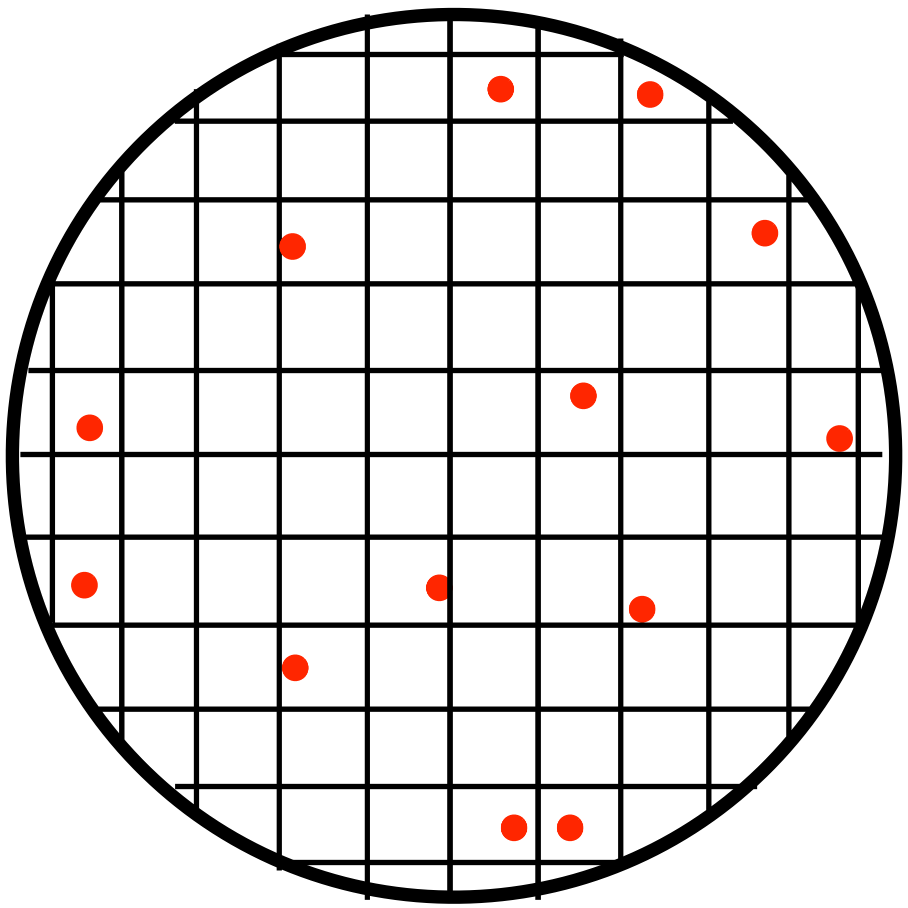
- Cost
- Energy
- Speed
- Reliability



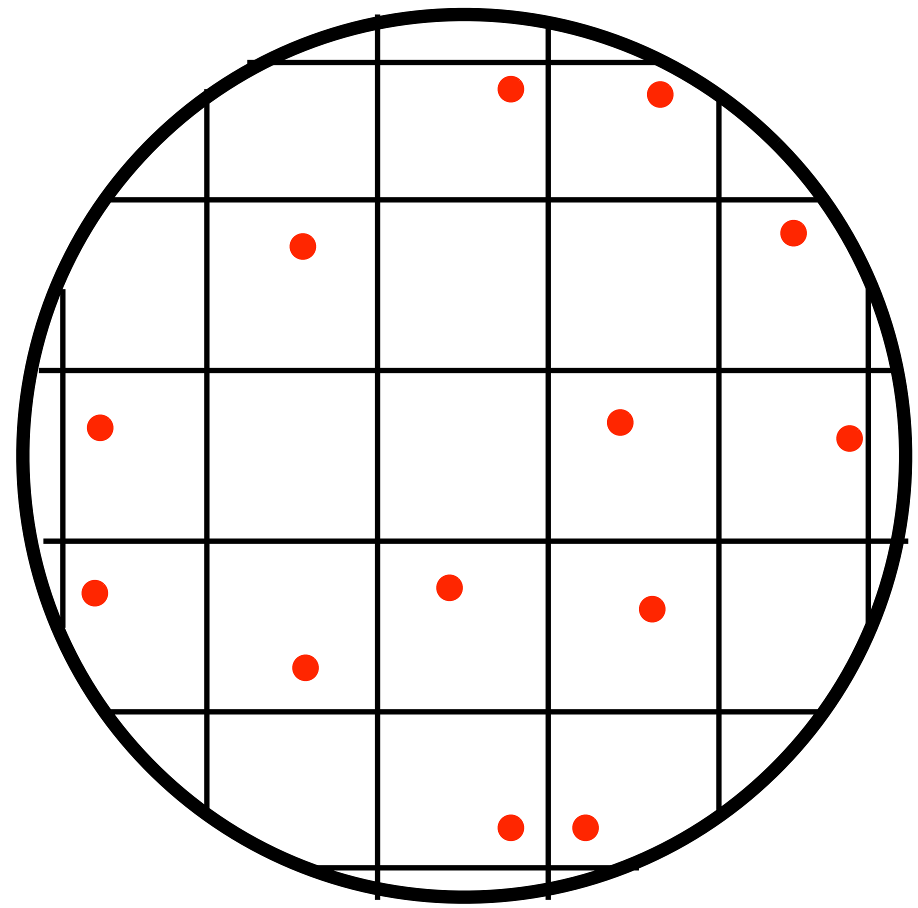
- Predictability / real-time (fast is *not* equal to real-time)

Number representation (e.g. fixed/floating-point, #bits) has a **major** impact on the design

Size is Very Important in Microprocessor Design



Die area = 1
Working = 64



Die area = 4
Working = 4

Cost per die = $f(area^{\boldsymbol{x}})$, $\boldsymbol{x} \in [2,4]$

Computational Resources for an Adder

Xilinx Virtex-7 XT 1140 FPGA:

Number representation	Registers (FFs)	Latency/delay (clock cycles)
double floating-point 52-bit mantissa	1046	14
single floating-point 23-bit mantissa	557	11
fixed-point 53 bits	53	1
fixed-point 24 bits	24	1

Cheap and **low power** processors often only have **fixed-point**

Dynamic Optimization

$$\min_{x(\cdot),u(\cdot),p} \; J(y(\cdot), x(\cdot), u(\cdot), p)$$

$$F(y(t), \dot{x}(t), x(t), u(t), p, t) = 0, \quad \forall t \in [t_0, t_f)$$
$$G(y(t), \dot{x}(t), x(t), u(t), p, t) \leq 0, \quad \forall t \in [t_0, t_f)$$

Discretized and approximated by finite-dimensional NLP:

$$\min_{\theta} V(\theta)$$

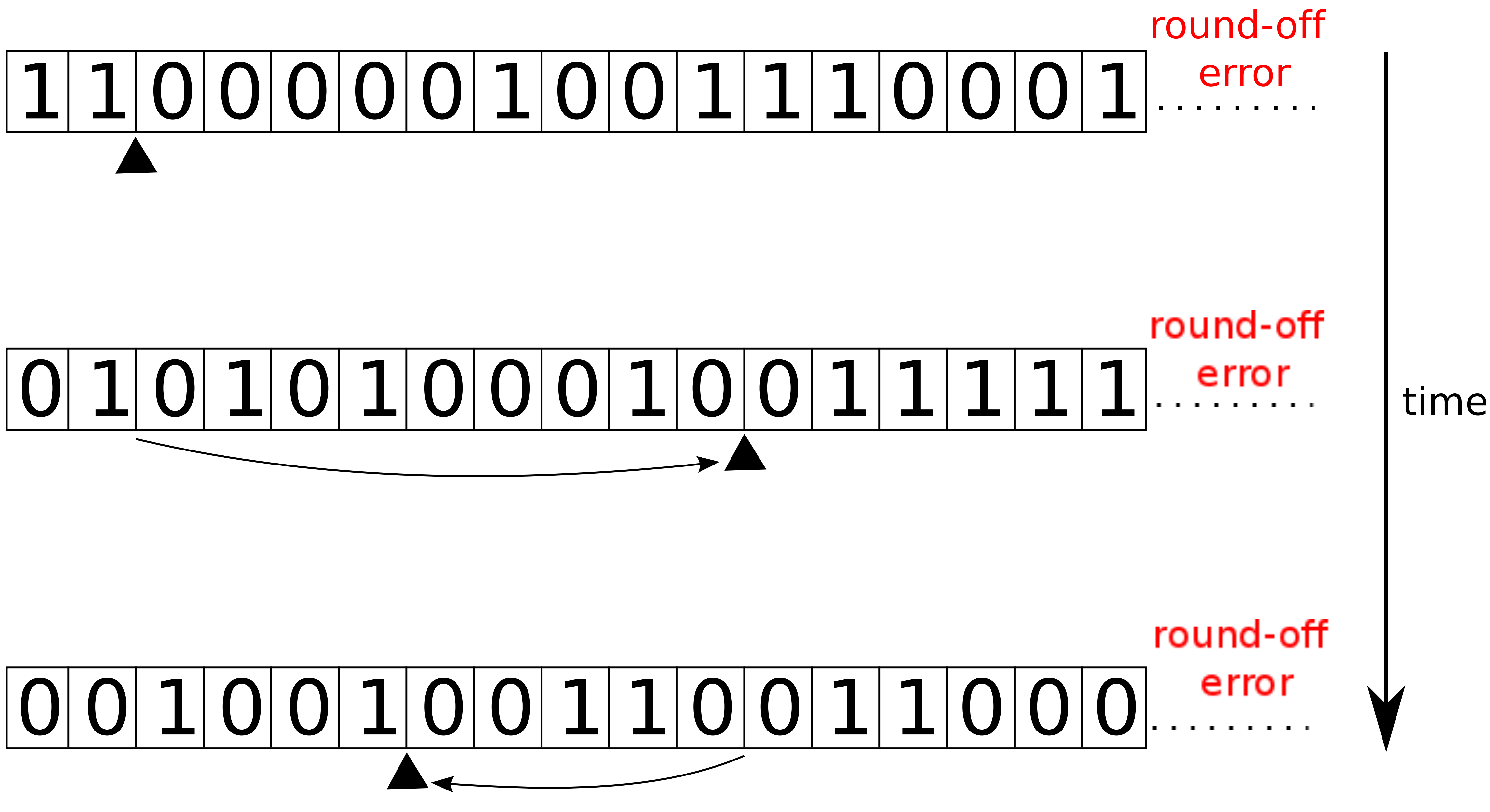
$$f(\theta) = 0$$

$$g(\theta) \leq 0$$

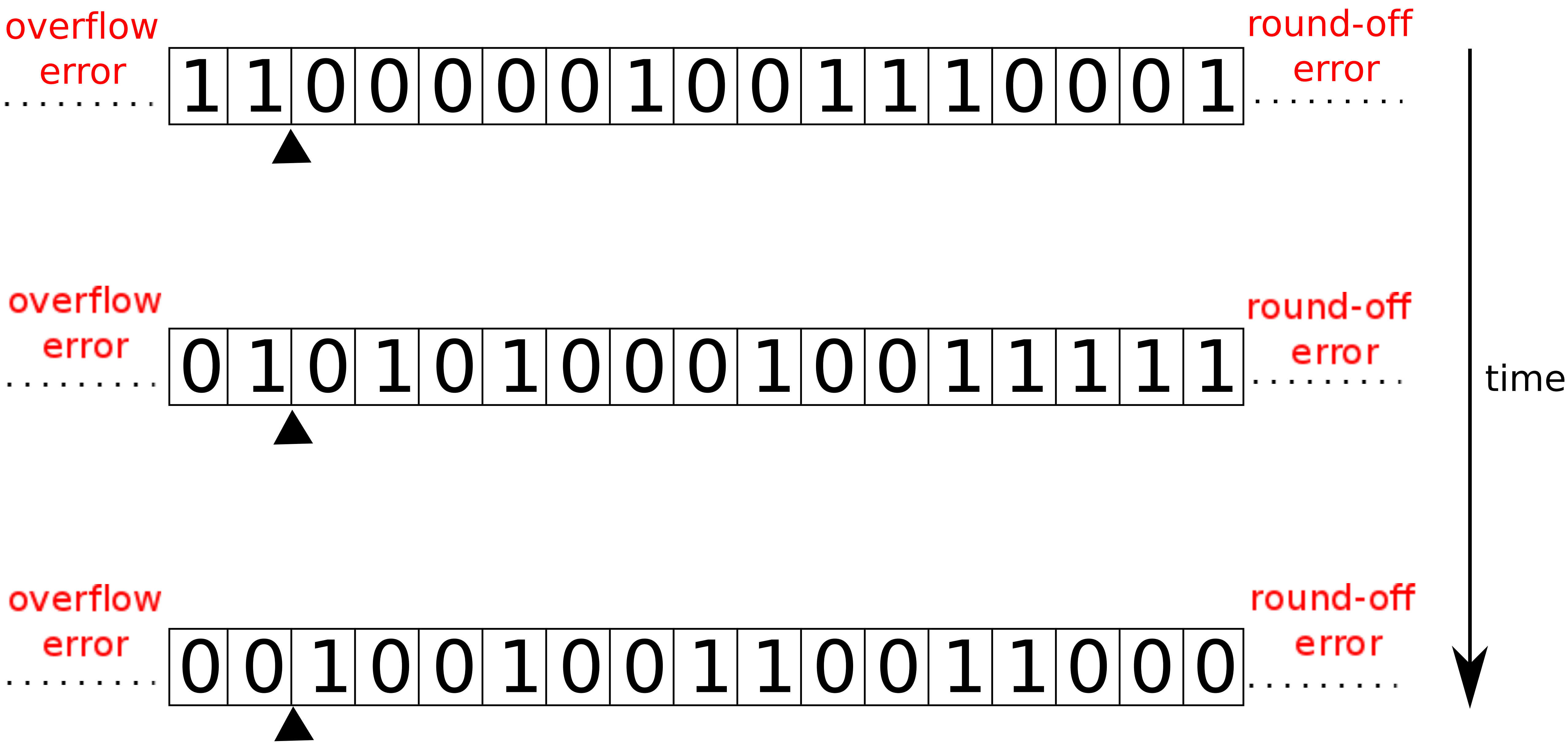
$$\theta \in \mathbb{R}^n, \;\; f : \mathbb{R}^n \rightarrow \mathbb{R}^m, \;\; g : \mathbb{R}^n \rightarrow \mathbb{R}^p$$

Fixed-Point Arithmetic

Floating-Point Arithmetic



Fixed-Point Arithmetic



Challenges for Fixed-Point Arithmetic

- Number of bits for **integer** and fractional part?
  - Determine **worst-case peak** values
  - Optimization algorithms are **nonlinear** and **recursive**
- **Search direction** most computationally critical part:

$$\mathsf{A}\xi = \mathsf{b}$$

- **Iterative linear solvers** preferred: CG, MINRES, GMRES

Lanczos Algorithm (Kernel of CG/MINRES)

$$\mathsf{Q}_i^T \mathsf{A} \mathsf{Q}_i = \mathsf{T}_i := \begin{bmatrix} \alpha_1 & \beta_1 & & 0 \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{i-1} \\ 0 & & \beta_{i-1} & \alpha_i \end{bmatrix}$$

Given q_1 such that $\|q_1\|_2 = 1$ and an initial value $\beta_0 := 1$

for $i = 1$ to i_{max} **do**

1. $q_i \leftarrow \frac{q_i}{\beta_{i-1}}$
2. $z_i \leftarrow \mathsf{A} q_i$
3. $\alpha_i \leftarrow q_i^T z_i$
4. $q_{i+1} \leftarrow z_i - \alpha q_i - \beta_{i-1} q_{i-1}$
5. $\beta_i \leftarrow \|q_{i+1}\|_2$

end for

Lanczos Algorithm (Kernel of CG/MINRES)

$$\mathsf{Q}_i^T \mathsf{A} \mathsf{Q}_i = \mathsf{T}_i := \begin{bmatrix} \alpha_1 & \beta_1 & & 0 \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{i-1} \\ 0 & & \beta_{i-1} & \alpha_i \end{bmatrix}$$

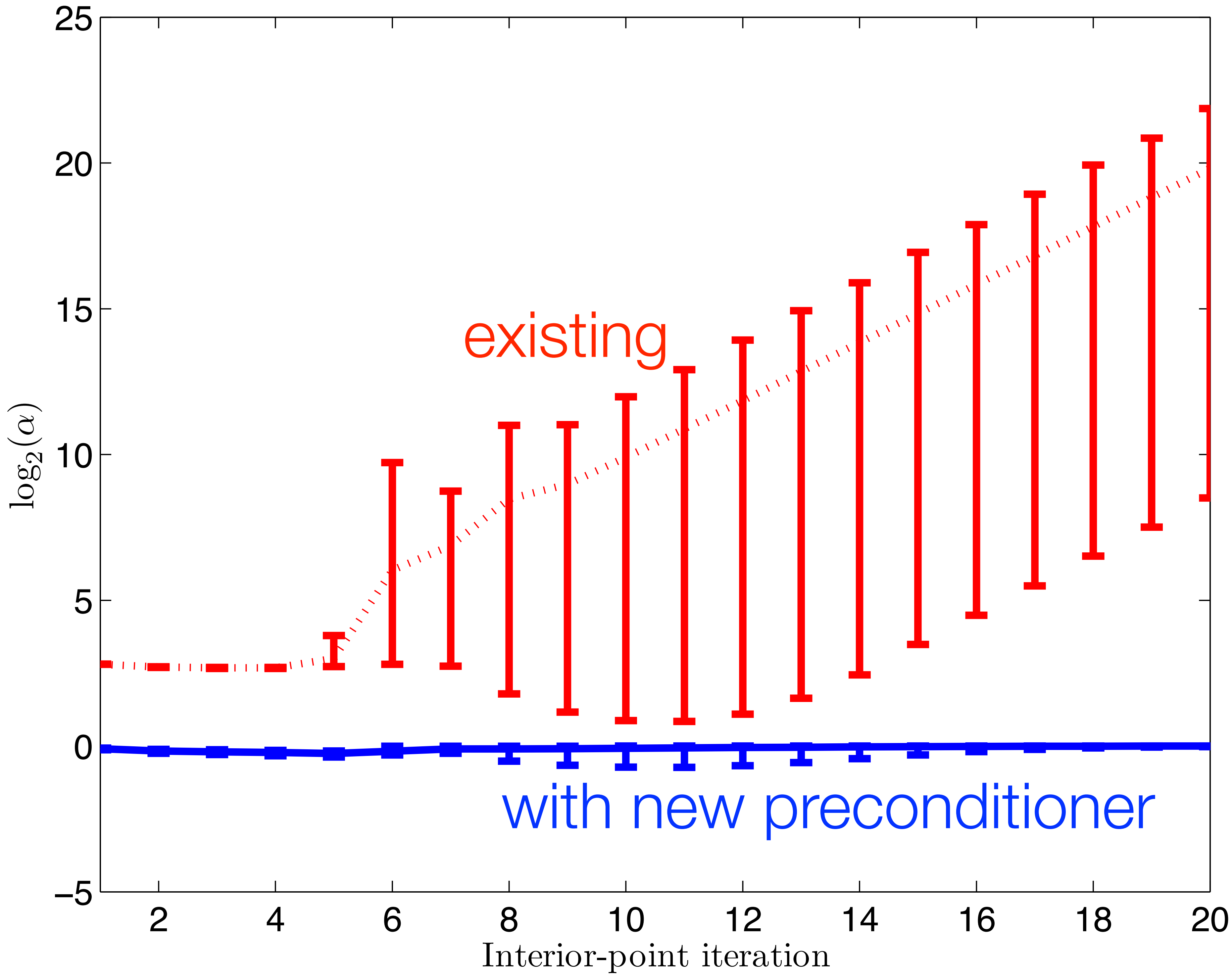
Given q_1 such that $\|q_1\|_2 = 1$ and an initial value $\beta_0 := 1$

for $i = 1$ to i_{max} **do**

1. $q_i \leftarrow \frac{q_i}{\beta_{i-1}}$
2. $z_i \leftarrow \mathsf{A} q_i$
3. $\alpha_i \leftarrow q_i^T z_i$
4. $q_{i+1} \leftarrow z_i - \alpha q_i - \beta_{i-1} q_{i-1}$
5. $\beta_i \leftarrow \|q_{i+1}\|_2$

end for

Evolution of Variables in Primal-dual Interior Point



Optimal control of a Boeing 747

On-line Diagonal Preconditioner / Scaler

$$\mathsf{A}\xi = \mathsf{b}, \quad \mathsf{A} = \mathsf{A}'$$

$$\mathsf{S}_{kk} := \sum_{j=1}^{\mathsf{N}} |\mathsf{A}_{kj}| \quad \text{(1-norm of row } k\text{)}$$

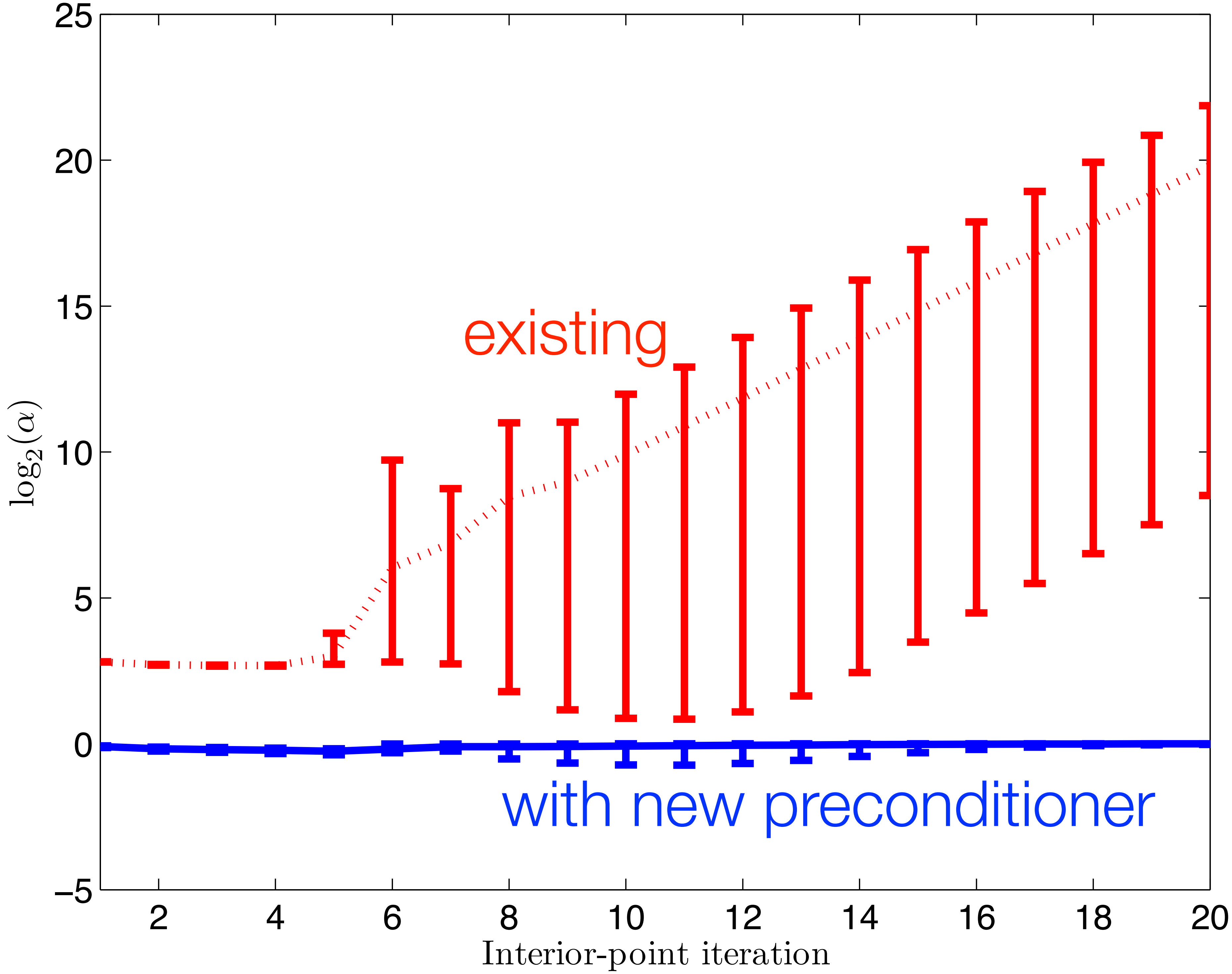
$$\mathsf{S}^{-\frac{1}{2}}\mathsf{A}\mathsf{S}^{-\frac{1}{2}}\psi = \mathsf{S}^{-\frac{1}{2}}\mathsf{b} \Leftrightarrow \widehat{\mathsf{A}}\psi = \widehat{\mathsf{b}} \Rightarrow \rho\left(\widehat{\mathsf{A}}\right) \leq 1$$

$$\xi = \mathsf{S}^{-\frac{1}{2}}\psi$$

Theorem (Avoiding overflow in fixed-point)
For integer part, need at most 2 bits for all the variables in the Lanczos algorithm and $\lceil -\log_2(\epsilon) \rceil$ bits for $1/\beta_i$

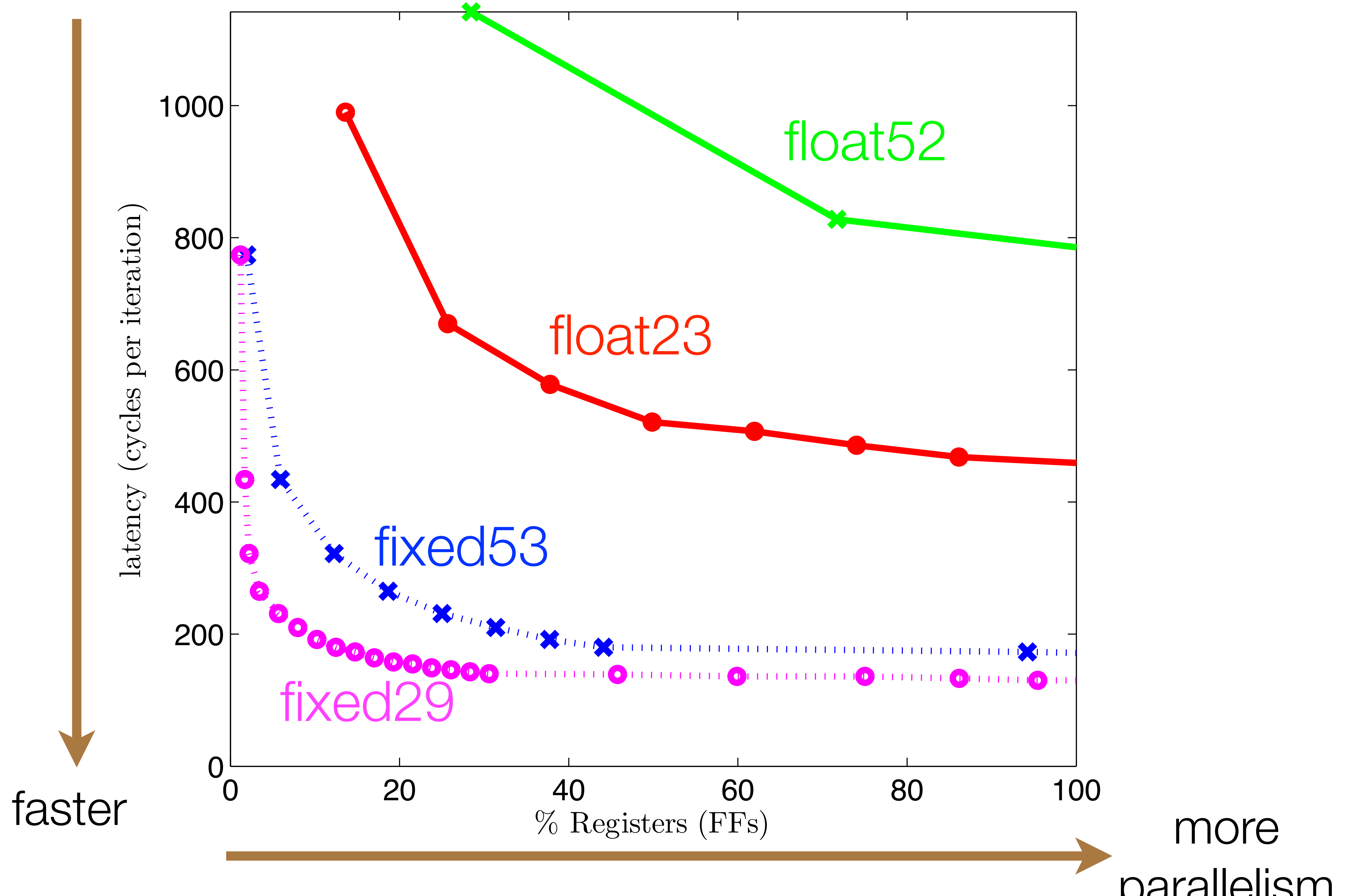
Proof: Proc. IEEE Conference on Decision and Control 2012

Evolution of Variables in Primal-dual Interior Point



Optimal control of a Boeing 747

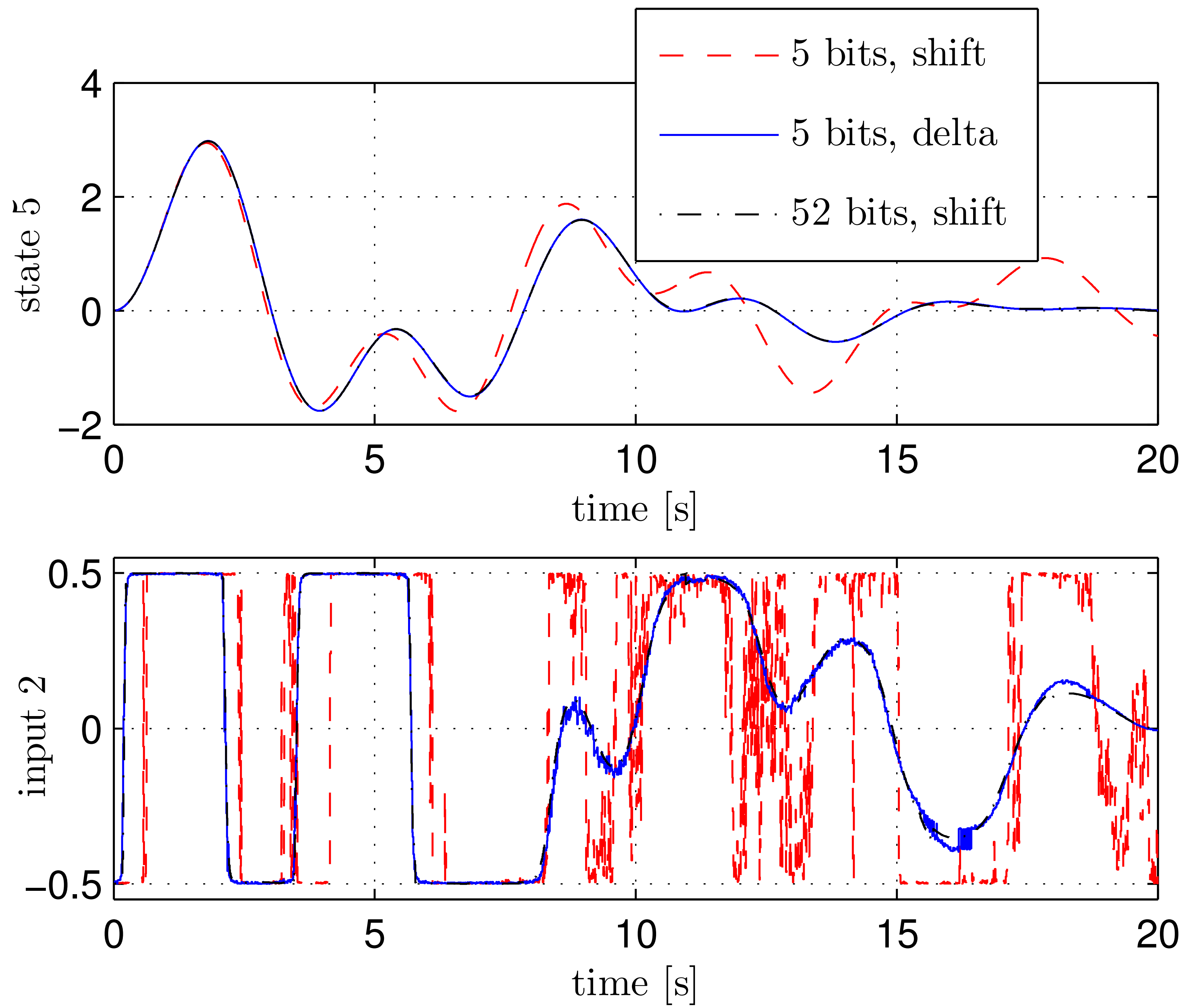
Trade-offs on an FPGA (same accuracy)



Xilinx Virtex-7 XT 1140 with matrices from the optimal control of a Boeing 747

Low-Precision Arithmetic

Optimal Control in Low Precision Arithmetic



Mass-spring system with 3 masses (6 states) and 2 inputs, sample period = 10ms

Sampled-data Representation in Shift Form

$$\dot{x}(t) = A_c x(t) + B_c u(t)$$

Sample period *h* and piecewise constant input (ZOH):

$$u(t) = u(kh) =: u_k, \quad \forall t \in [kh, kh+h)$$

Exact solution/discrete-time model to compute $x_k := x(kh)$

$$x_{k+1} = A_s x_k + B_s u_k$$

$$A_s := e^{A_c h} = I + A_c h + \frac{(A_c h)^2}{2!} + \frac{(A_c h)^3}{3!} + \ldots$$

$$\lim_{\|A_c h\| \to 0} A_s = I, \qquad \lim_{\|A_c h\| \to 0} B_s = 0$$

Sampled-data Representation in Shift Form

$$\dot{x}(t) = A_c x(t) + B_c u(t)$$

Sample period *h* and piecewise constant input (ZOH):

$$u(t) = u(kh) =: u_k, \quad \forall t \in [kh, kh + h)$$

Exact solution/discrete-time model to compute $x_k := x(kh)$

$$\boxed{x_{k+1} = A_s x_k + B_s u_k}$$

$$A_s := e^{A_c h} = I + A_c h + \frac{(A_c h)^2}{2!} + \frac{(A_c h)^3}{3!} + \ldots$$

$$\lim_{\|A_c h\| \to 0} A_s = I, \qquad \lim_{\|A_c h\| \to 0} B_s = 0$$

Sampled-data Representation in Shift Form

$$\dot{x}(t) = A_c x(t) + B_c u(t)$$

Sample period *h* and piecewise constant input (ZOH):

$$u(t) = u(kh) =: u_k, \quad \forall t \in [kh, kh+h)$$

Exact solution/discrete-time model to compute $x_k := x(kh)$

$$\boxed{(x_{k+1} - x_k)/h = (A_s x_k + B_s u_k - x_k)/h}$$

$$A_s := e^{A_c h} = I + A_c h + \frac{(A_c h)^2}{2!} + \frac{(A_c h)^3}{3!} + \dots$$

$$\lim_{\|A_c h\| \to 0} A_s = I, \qquad \lim_{\|A_c h\| \to 0} B_s = 0$$

Sampled-data Representation in Delta Form

Middleton and Goodwin (IEEE TAC, 1986):

$$\frac{x_{k+1} - x_k}{h} = \frac{(A_s - I)}{h} x_k + \frac{B_s}{h} u_k$$

Sampled-data Representation in Delta Form

Middleton and Goodwin (IEEE TAC, 1986):

$$\frac{x_{k+1} - x_k}{h} = A_\delta x_k + B_\delta u_k$$

Sampled-data Representation in Delta Form

Middleton and Goodwin (IEEE TAC, 1986):

$$\frac{x_{k+1} - x_k}{h} = A_\delta x_k + B_\delta u_k$$

$$A_\delta = A_c + \frac{A_c^2 h}{2!} + \frac{A_c^3 h^2}{3!} + \ldots$$

$$\lim_{\|A_c h\| \to 0} A_\delta = A_c, \qquad \lim_{\|A_c h\| \to 0} B_\delta = B_c$$

Equivalent to shift form in **infinite** precision arithmetic
Different from shift form in **finite** precision arithmetic

Optimization Problem Using Shift Form

$$\min_{\theta} \sum_{k=0}^{N-1} \ell_k(x_k, u_k)$$

$$\theta := \begin{bmatrix} u_0' & x_1' & u_1' & x_2' & \cdots & u_{N-1}' & x_N' \end{bmatrix}'$$

subject to

$$\begin{aligned} x_0 &= \hat{x}, \\ \boxed{x_{k+1} = A_s x_k + B_s u_k,} &\quad \forall k \in \{0, 1, \dots, N-1\} \\ J x_k + E u_k \leq d, &\quad \forall k \in \{0, 1, \dots, N-1\} \end{aligned}$$

Optimization Problem Using Shift Form

$$\min_{\theta} \sum_{k=0}^{N-1} \ell_k(x_k, u_k)$$

$$\theta := \begin{bmatrix} u_0' & x_1' & u_1' & x_2' & \cdots & u_{N-1}' & x_N' \end{bmatrix}'$$

subject to

$$\begin{aligned} x_0 &= \hat{x}, \\ \boxed{x_{k+1} = A_s x_k + B_s u_k,} & \quad \forall k \in \{0, 1, \ldots, N-1\} \\ J x_k + E u_k \leq d, & \quad \forall k \in \{0, 1, \ldots, N-1\} \end{aligned}$$

$$\boxed{\delta_k := \frac{x_{k+1} - x_k}{h} = A_\delta x_k + B_\delta u_k}$$

Optimization Problem Using Delta Form

$$\min_{\theta} \sum_{k=0}^{N-1} \ell_k(x_k, u_k)$$

$$\theta := \begin{bmatrix} u_0' & \boxed{\delta_0'} & x_1' & u_1' & \boxed{\delta_1'} & x_2' & \cdots & u_{N-1}' & \boxed{\delta_{N-1}'} & x_N' \end{bmatrix}'$$

subject to

$$\begin{aligned} x_0 &= \hat{x}, & \\ \delta_k &= A_\delta x_k + B_\delta u_k, & \forall k \in \{0, 1, \ldots, N-1\} \\ x_{k+1} &= x_k + h\delta_k, & \forall k \in \{0, 1, \ldots, N-1\} \\ Jx_k + Eu_k &\leq d, & \forall k \in \{0, 1, \ldots, N-1\} \end{aligned}$$

$$\boxed{\delta_k := \frac{x_{k+1} - x_k}{h} = A_\delta x_k + B_\delta u_k}$$

Optimization Problem Using Delta Form

$$\min_{\theta} \sum_{k=0}^{N-1} \ell_k(x_k, u_k)$$

$$\theta := \begin{bmatrix} u_0' & \boxed{\delta_0'} & x_1' & u_1' & \boxed{\delta_1'} & x_2' & \cdots & u_{N-1}' & \boxed{\delta_{N-1}'} & x_N' \end{bmatrix}'$$

subject to

$$\begin{aligned} x_0 &= \hat{x}, & \\ \delta_k &= A_\delta x_k + B_\delta u_k, & \forall k \in \{0, 1, \ldots, N-1\} \\ x_{k+1} &= x_k + h\delta_k, & \forall k \in \{0, 1, \ldots, N-1\} \\ Jx_k + Eu_k &\leq d, & \forall k \in \{0, 1, \ldots, N-1\} \end{aligned}$$

Solving the Optimization Problem

- Solve linearized **KKT system** (Rao, Wright, Rawlings; JOTA, 1998):

$$\mathsf{A}\xi = \mathsf{b}$$

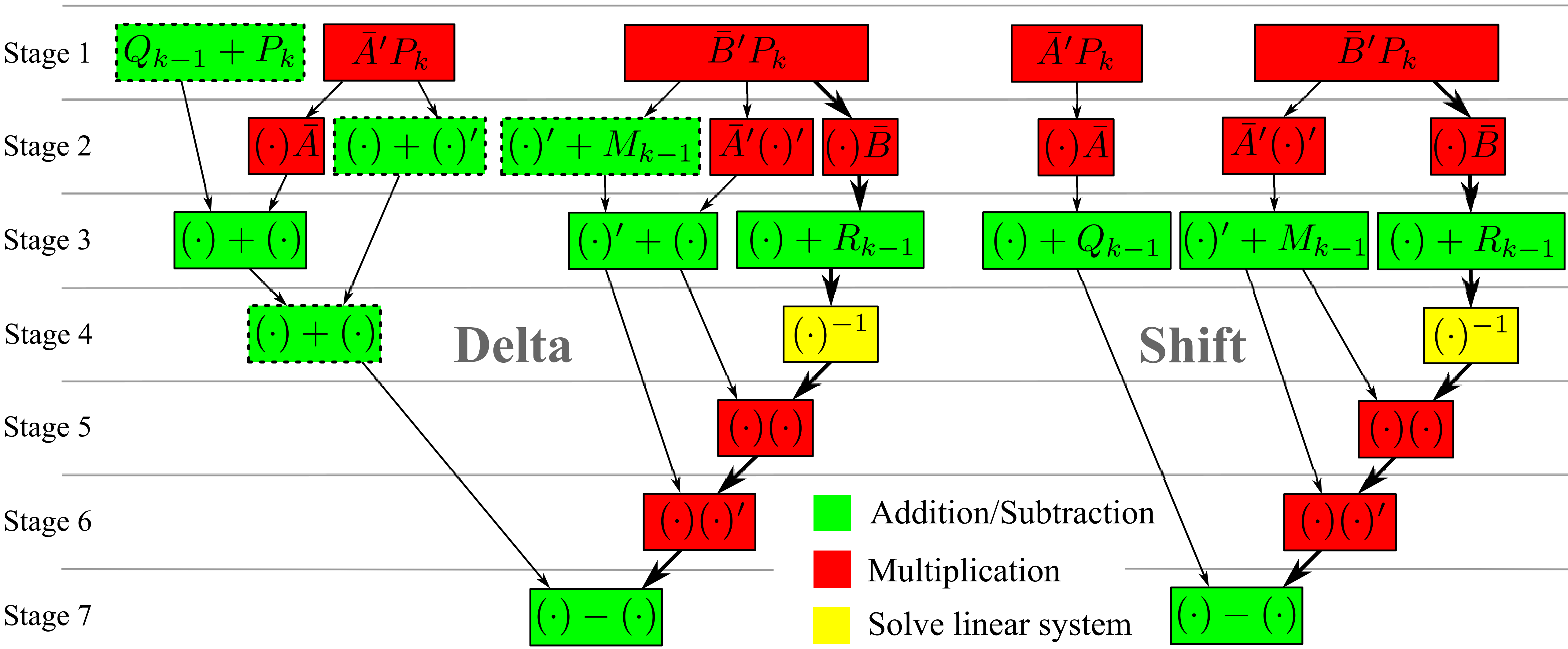
- **Interleave** search direction variables:

$$\xi := \begin{bmatrix} \Delta u_0' & \Delta \gamma_0' & \Delta \delta_0' & \Delta \lambda_1' & \Delta x_1' & \cdots & \Delta x_N' \end{bmatrix}'$$

- **Block elimination** results in Riccati recursions:

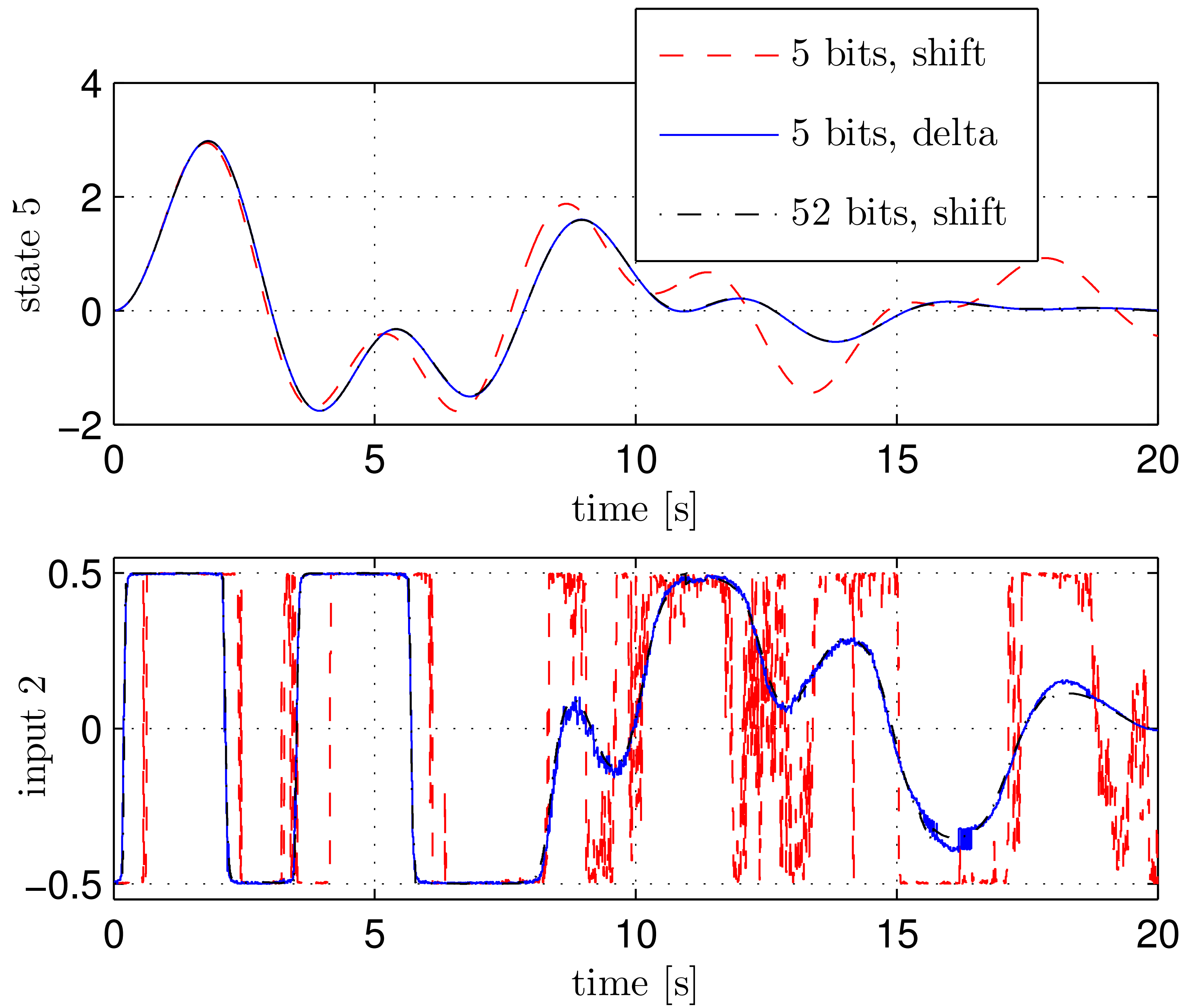
$$\begin{aligned} P_{k-1} := Q_{k-1} + P_k + h^2 A_\delta' P_k A_\delta + h A_\delta' P_k + h P_k A_\delta \\ -(M_{k-1} + h^2 A_\delta' P_k B_\delta + h P_k B_\delta)(R_{k-1} + h^2 B_\delta' P_k B_\delta)^{-1} \\ (M_{k-1}' + h^2 B_\delta' P_k A_\delta + h B_\delta' P_k) \end{aligned}$$

Data Dependencies in Riccati Recursion



Same amount of multipliers, adders and computational delay for a custom circuit, e.g. FPGA

Optimal Control in Low Precision Arithmetic



Mass-spring system with 3 masses (6 states) and 2 inputs, sample period = 10ms

Conclusions

- Number representation major factor that determines cost, energy, computational delay and accuracy

- Fixed-point: **Precondition** to get tight **analytical bounds** on variables in **Lanczos** algorithm to avoid **overflow**

- Low precision: **Sampled-data model** and **optimization method** crucial to successful implementation

- **Co-design** algorithm and hardware to use “just the right amount” of computational resources

Open Research Questions

- Other sampled-data and number representations?
- Nonlinear dynamical models?
- Which algorithms map easily to low precision, fixed-point or other number representations?
- A priori guarantees on accuracy, closed-loop stability, robustness and performance?
- Need control + optimization + numerics + computing

Designing for Embedded Systems

"All too often, today's students use **laptop** [or **desktop**] computers to perform their computing, which shields them from dealing with any of the physical **constraints** they will face in the **real world**. This approach is akin to **trying to learn skiing while standing comfortably in the après ski lounge**."

Wolf, Cyber-Physical Systems, *Computer*, 2009.